

Computer Technology

Parallel Processing System

In the latter 1970s, NASA saw a need for computing power orders of magnitude beyond anything then available for satellite image analysis. Already operating at that time were satellites relaying voluminous information to Earth at high transmission rates, such as NASA's Earthscanning Landsat resources survey satellite sending digital data to ground stations at the rate of 15 million bits per second. On the developmental horizon were satellites of far greater data gathering and transmission capacities.

To provide a

image data from

capability for processing very high resolution

processing
involves
simultaneous
processing of
image

picture

Parallel

elements

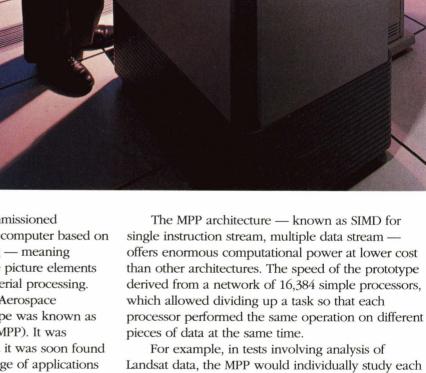
rather than

step-by-step

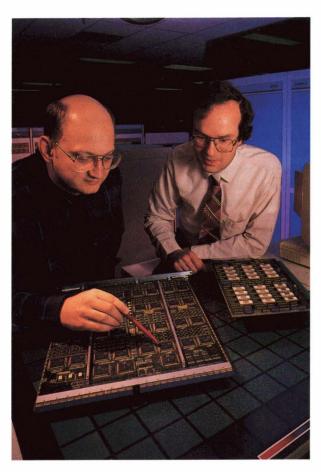
serial

processing

spacecraft sensors,
Goddard Space Flight Center commissioned
development of a unique type of computer based on
the concept of parallel processing — meaning
simultaneous processing of image picture elements
(pixels) rather than step-by-step serial processing.
Designed and built by Goodyear Aerospace
Corporation, the resulting prototype was known as
the Massively Parallel Processor (MPP). It was
delivered to Goddard in 1983 and it was soon found
to have utility in a far broader range of applications
than just image processing.



For example, in tests involving analysis of Landsat data, the MPP would individually study each of the million pixels in a typical image and decide whether each dot represented land or water, forest or farmland, or whatever else. In other words, in massively parallel processing an entire image is processed at once, where in serial processing an image is processed one pixel at a time; the latter takes hours to analyze and classify an image, MPP about 20 seconds.



In order to measure and document the advantages and disadvantages of parallel processing, and to learn the capabilities and limitations of the MPP, NASA organized a working group of 40 scientists who were provided opportunities to test their computational algorithms on the MPP beginning in the fall of 1985.

A year later, sufficient results had been acquired to warrant convening — at Goddard — the first symposium on massively parallel scientific computation. The MPP investigators described a broad variety of applications, including signal and image processing, Earth science, physical science, computer science and graphics. The performance of many of these applications was found to be in the supercomputer range, and for certain tasks MPP was found to be faster than traditional vector supercomputers. Subsequently, Goddard funded the

Microelectronics Center of North Carolina's development of a second generation MPP called the Blitzen Project to demonstrate that the size and weight of the MPP could be reduced enough to allow its use in spacecraft.

Based in part on technology developed in the two NASA MPP projects, MasPar Computer Corporation, Sunnyvale, California, has produced a new generation massively parallel computing system. The MasPar MP-1 product family ranges upward from a unit with 1,024 processors that can deliver 1,600 MIPS (millions of instructions per second) and 82 MFLOPS (millions of floating operations per second). At the upper end of the scale, with 16,384 processors, the MP-1 can deliver 26,000 MIPS and 1,300 MFLOPS. MP-1 users, including NASA, are attacking computationally-intensive problems in such areas as image processing and understanding, signal processing, database management query systems, neural-net algorithms, computational fluid dynamics, content-sensitive text retrieval and seismic data reduction.

At far left, James R. Fischer, head of Goddard's image analysis facility, is shown with the MasPar MP-1 (foreground) and the earlier MPP prototype (blue unit in background); the photo emphasizes the compactness of the new system as compared with its predecessor. Fischer (white shirt) and a colleague are shown at left checking out one of the MP-1's circuit boards.

NASA
research led
to a new
generation
massively
parallel
computing
system